Jurnal Media Aplikom
ISSN : 2086-972 XX
Vol. 4 No. 3 September 2015

SEGMENTASI CITRA AKSARA JAWA MENGGUNAKAN ALGORITMA PARTICLE SWARM OPTIMIZATION PARAREL

Oskar Ika Adi Nugroho
Dosen Sekolah Tinggi Ilmu Komputer Yos Sudarso

ABSTRAK

Aksara Jawa merupakan ciri khas dan warisan leluhur dari Suku Jawa yang perlu dijaga dan dilestarikan keberadaannya. Penulis mengembangkan perangkat lunak proses segmentasi citra aksara jawa. Tujuan dari dilakukan segmentasi citra adalah mengubah representasi dari suatu citra menjadi sesuatu yang lebih berarti dan mudah untuk dianalisis . Segmentasi citra dengan menggunakan metode *clustering* dapat digunakan dengan berbagai macam metode, salah satunya adalah P*article Swarm Optimization* (PSO). Dalam prosesnya digunakan metode *Particle Swarm Optimization* (PSO) untuk proses segmentasi pada karakter Aksara Jawa. Untuk mengatasi beban komputasi yang cukup berat pada proses segmentasi citra, maka diterapakan metode *parallel programing*, dan arsitektur komputer yaitu GPU dengan support CUDA dari NVIDIA. Hasil akhir dari penelitian ini dapat mengimplementasikan segmentasi citra digital dengan metode *Particle Swarm Optimization* pada Aksara Jawa. GPU CUDA mempercepat proses segmentasi citra Aksara Jawa

Keywords: aksara jawa, *segmentation*, *particle swarm optimization*, *parallel programing*, GPU, CUDA.

Jurnal Media Aplikom
ISSN : 2086-972 XX
Vol. 4 No. 3 September 2015

I. PENDAHULUAN

Aksara Jawa yang menjadi bagian tak terpisahkan dari Bahasa Jawa dan merupakan salah satu unsur kebudayaan dari masyarakat Jawa, saat ini tinggal mengalami kematian. Pengunaan Aksara Jawa semakin berkurang. Jumlah website yang menggunakan Bahasa Jawa sangat sedikit, padahal ada sekitar 80.000.000 pengguna Bahasa Jawa sebagai bahasa sehari hari dan Bahasa Jawa mempunyai sejarah penulisan hampir 1200 tahun. Hanya ditemukan sekitar 45 website menggunakan Bahasa Jawa, demikian juga e-group paling besar dan paling aktif hanya memiliki 126 anggota berdasarkan data 18 Februari 2002.(Arps & Supriyanto, 2002).

Tujuan dari dilakukan segmentasi citra adalah mengubah representasi dari suatu citra menjadi sesuatu yang lebih berarti dan mudah untuk dianalisis. Segmentasi citra dengan menggunakan metode clustering dapat digunakan dengan berbagai macam algoritma, salah satunya adalah Particle Swarm Optimization (PSO). Particle Swarm Optimization (PSO) yang telah dimodifikasi dapat digunakan untuk melakukan clustering pada citra digital. (Kristiadi, Pranowo, & Mudjihartono, 2013)

II. TINJAUAN PUSTAKA

Segmentasi Citra Teks Sastra Jawa, menggunakan metode profil proyeksi telah dilakukan (Widiarti, 2007) dan tingkat kesuksesan 86,78% mensegmentasi dokumen teks Sastra Jawa. Particle Swarm Optimization telah digunakan untuk mengatasi berbagai masalah masalah pada kajian segmentasi dan pengenalan pola (Merwe & Engelbrecht, 2003)(Cheo & Ye, 2004)(Omran M. , 2004)(Omran, Engelbrecht, & Salman, 2005)(Cui, Potok, & Palathingal, Document Clustering using Particle Swarm Optimization, 2005)(Cui & Potok, Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm, 2005)(Abraham, Das, & Roy, 2008)(Murugesan & Palaniswami, 2010)(Mohsen, Hadhoud, & Amin, 2011)(Yih, Lin, & Liu, Clustering Analysis Method based on Fuzzy C-Means Algorithm of PSO and PPSO with Application in Real Data, 2007)(Yih, Lin, & Liu, Clustering Analysis Method based on Fuzzy C-Means Algorithm of PSO and

PPSO with Application in Image Data, 2008)(Ye & Chen, 2005)(Lin, Wang, & Lee, 2009)(Santosa & Ningrum, 2009). Particle Swarm Optimization bisa melakukan segmentasi citra dengan hasil yang memuaskan (LAI , 2006). PSO memberikan kinerja yang menjanjikan dan perilaku yang stabil dalam mengenali angka tulisan tangan(Ba-Karait & Shamsuddin, 2008).

Penggunaan teknologi *GPU* banyak diterapkan juga pada algoritma *Particle Swarm Optimization* (Mussi, et al., 2009) (Zhou & Tan, 2009) (Zhou & Tan, 2011). Jika menggunakan NVIDIA CUDA untuk mempararelkan komputasi, PSO berjalan lebih cepat 170% pada *GPU*, daripada berjalan pada *CPU* dengan jumlah partikel 100. (Kristiadi, Pranowo, & Mudjihartono, PARALLEL PARTICLE SWARM OPTIMIZATION FOR IMAGE SEGMENTATION, 2013). Pada permasalahan konvergensi untuk optimalisasi fungsi *Rastrigin* dan fungsi *Ackley* di ruang pencarian 30-dimensi dengan menggunakan *General Purpose Graphics Processing Unit* (*GPGPU*) untuk pemrosesan paralel dari PSO, fungsi *Rastrigin* berjalan lebih cepat 16 kali dari komputer *cluster* dan untuk masalah *Ackley* itu lebih cepat 8 kali daripada komputer *cluster*. (Liera, et al., 2011). Teknik yang didasarkan pada metode *Sauvola-Pietkinen* untuk mendeteksi tepi dengan kesinambungan menunjukkan bahwa algoritma PSO yang memanfaatkan teknik *thresholding* lokal baru, lebih baik daripada yang menggunakan metode *Otsu* (Setayesh, et al., 2009). Masalah estimasi parameter yang realistis di mana setiap prosesor melakukan perhitungan yang signifikan dengan metode PSO berhasil menunjukkan peningkatan percepatan ditandai dengan ukuran populasi (Wachowiak & Foster, 2012). Implementasi *clustering K-Means* secara pararel telah dilakukan (Farivar, et al., 2008) menghasilkan peningkatan kinerja 13kali.

III. LANDASAN TEORI

Aksara Jawa

Masyarakat Jawa sudah mempunyai bentuk penulisan aksara, yang dianggap adiluhung leluhur bangsa Jawa hingga kini. Aksara Jawa yang menjadi bagian tak terpisahkan dari bahasa Jawa dan merupakan salah satu unsur kebudayaan dari masyarakat Jawa. Perkembangan Aksara Jawa juga ada kaitan dengan perkembangan bahasa Jawa yang lahir sebagai alat

komunikasi masyarakat. Namun, Aksara Jawa dipercayai muncul setelah berlaku pertemuan antara peradaban Jawa dengan India. Sebelum itu, tidak terdapat bukti yang menunjukkan bangsa Jawa mempunyai aksaranya sendiri. Aksara Jawa yang ada pada masa kini adalah hasil daripada pembentukan kembali bentuk dan gaya Aksara Jawa Kuna (kuno). (Mohamed, 2001)

Aksara Jawa merupakan turunan Aksara Brahmi sebagaimana Aksara Nusantara lainnya, memiliki kedekatan dengan Aksara Bali. Aksara Jawa memiliki bentuk yang lebih rumit dibandingkan dengan Aksara Latin biasa. Aksara Jawa terdiri atas aksara dasar yang disebut dengan Aksara *Carakan* atau *nglegeno* atau Aksara Jawa tanpa *sandangan*, Aksara *Pasangan*, Aksara *Swara*, Aksara *Rekan*, Aksara *Murda*, *Wilangan* atau Angka, dan *Sandangan* atau tanda baca (Nugraha, 2009).

Parallel computing

Komputasi paralel merupakan salah satu teknik melakukan komputasi secara bersamaan dengan memanfaatkan beberapa komputer juga secara bersamaan. Pada komputasi paralel dibutuhkan kapasitas yang sangat besar untuk memproses komputasi yang banyak. Di samping itu pemakai harus membuat pemrograman paralel untuk dapat merealisasikan komputasi. (Sanders & Kandrot, 2011)

Pemrograman paralel memiki tujuan utama yaitu untuk meningkatkan performa komputasi karena semakin banyak hal yang bisa dilakukan secara bersamaan dalam waktu yang sama, maka semakin banyak pula pekerjaan yang bisa diselesaikan (Kirk & Hwu, 2010).

Stream Programming adalah sebuah teknik pemrograman konkruen yang dapat digunakan untuk melakukan komputasi paralel. Dalam *stream programming*, data diorganisasi menjadi sekumpulan *stream*, dan sebuah program yang disebut *kernel* diaplikasikan pada setiap elemen *stream*. *Stream programming* menggunakan paradigma *Single Instruction Multiple Data* (SIMD). Artinya, program yang sama dieksekusi pada sekumpulan data secara paralel. (Kirk & Hwu, 2010)

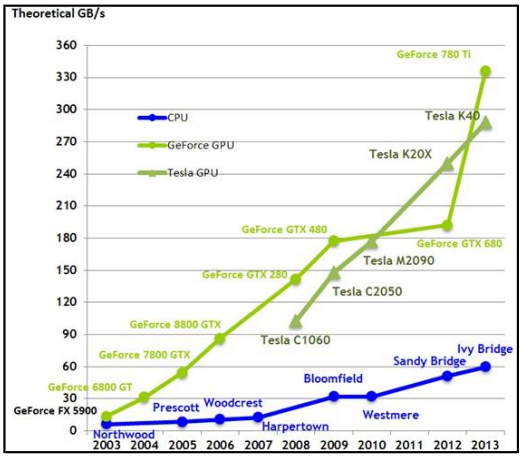
Salah satu perangkat keras yang dapat digunakan untuk melakukan stream programming adalah *Graphics Processing Unit* (GPU). GPU sangat cocok untuk diprogram dengan teknik ini, karena GPU

dirancang dari awal untuk melakukan proses *rendering* 3D, dimana proses tersebut membutuhkan *hardware* yang dirancang secara khusus untuk memproses data secara SIMD. Untuk mengimplementasikan stream programming pada GPU, ada banyak framework yang dapat dipakai. Salah satunya adalah *Compute Unified Device Architecture* (CUDA). (CUDA™, 2012)

Kemampuan Komputasi GPU

Graphic Processing Unit (GPU) adalah salah satu *peripheral* komputer yang bertugas untuk melakukan proses *rendering* objek 3D ke layar. Karena *rendering* merupakan proses yang sangat paralel, maka evolusi arsitektur GPU pun mengikutinya: GPU berevolusi menjadi sebuah multiprosesor yang mampu menjalankan tugas secara paralel.

GPU dapat disebut unit komputasi yang terpisah dari CPU karena GPU memiliki prosesor dan memori sendiri. GPU memiliki jumlah *Arithmetic Logic Unit* (ALU) yang jauh lebih besar dibandingkan komputer biasa. Oleh karena itu, kemampuan GPU dalam mengolah data yang besar dapat diandalkan. Pada gambar 2.3, dapat dilihat bahwa pertumbuhan kemampuan GPU jika diukur dengan *Floating Point Operations per Second* (FLOPS) jauh meninggalkan CPU, bahkan meninggalkan hukum Moore.



Gambar 1. Peningkatan kecepatan GPU

Compute Unified Device Architecture (CUDA)

CUDA™ adalah platform komputasi paralel dan model pemrograman yang memungkinkan peningkatan dramatis dalam kinerja komputasi dengan memanfaatkan kekuatan dari graphics processing unit (GPU). (Kirk & Hwu, 2010) Sejak diperkenalkan pada 2006, CUDA telah banyak disebarkan melalui ribuan aplikasi dan diterbitkan makalah penelitian, dan terinstal lebih dari 300 juta CUDA-enabled GPU di notebook, workstation, komputer cluster dan super-komputer. GPU diaplikasikan juga untuk astronomi, biologi, kimia, fisika, data mining, manufaktur, keuangan. (Sanders & Kandrot, 2011)

Clustering

Clustering merupakan proses untuk mengidentifikasi pengelompokan di dalam multidimensi berdasarkan suatu kemiripan tertentu. *Clustering* banyak digunakan untuk penambangan data, analisis data statistik, analisis citra, pengenalan pola, *information retrieval*, dan bioinformatika (Merwe & Engelbrecht, 2003).

Kualitas suatu algoritma *clustering* ditentukan oleh dua faktor, yaitu kemiripan suatu data yang ada di dalam suatu *cluster* yang sama dan ketidakmiripan suatu data dengan data lain yang ada pada *cluster* yang berbeda. Semakin mirip data yang ada dari dalam suatu *cluster* yang sama, maka kualitas algoritma tersebut semakin baik. Semakin tidak mirip suatu data dengan data lain yang ada di *cluster* yang berbeda, maka kualitas algoritma tersebut semakin baik. Ukuran kualitas algoritma *clustering* yang biasa digunakan adalah *Quantization Error* (Omran, Engelbrecht, & Salman, 2005).

Particle Swarm Optimization

Particle Swarm Optimization adalah algoritma optimasi sederhana namun *powerful*, diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995 (Mussi, Cagnoni, & Daolio, 2009). Pencarian *PSO* untuk optima dari suatu fungsi, disebut fungsi *fitness*, mengikuti aturan terinspirasi oleh perilaku kawanan burung mencari makanan. Algoritma ini memodelkan populasi dari *swarm* sebagai titik pencari yang bergerak secara stokastik pada ruang pencarian. Posisi terbaik dari suatu individu dalam *swarm* tersebut disimpan, dan dinamakan dengan pengalaman (*experience*) dari pertikel tersebut. Pengalaman pengalaman dari suatu partikel pada *swarm* dikomunikasikan ke semua partikel yang ada, sehingga gerakan gerakan partikel pada *swarm* akan cenderung menuju ke suatu arah berdasarkan pengalaman pengalaman yang ada.

Gambar 2. Perilaku Kawanan Burung

Algoritma ini dimulai dengan membuat himpunan partikel partikel secara acak yang dinamakan dengan *swarm*. Kemudian setiap partikel tersebut mengkalkulasi kecepatan dan posisi barunya untuk setiap dimensi yang ada pada ruang pencarian. Kecepatan partikel dipengaruhi

oleh 3 faktor, yaitu kecepatan saat itu, pengalaman partikel tersebut, dan posisi terbaik dari semua partikel yang ada pada *swarm*. Sehingga untuk partikel *i*, pada dimensi *d*, kecepatannya dapat dikalkulasi dengan persamaan berikut

$$V_{id} = W * V_{id} + c_1 * r_1 * \left(P_{id} - X_{id}\right) + c_2 * r_2 * \left(P_{gd} - X_{id}\right) \quad (1)$$

Dimana W adalah *inertia weight* yang digunakan untuk menyeimbangkan kemampuan partikel dalam mencari solusi optimal global dan lokal, c_1 dan c_2 merupakan konstanta non negatif. r_1 dan r_2 merupakan bilangan acak diantara 0 dan 1. P_{id} merupakan posisi terbaik partikel tersebut berdasarkan pengalamannya, P_{gd} merupakan posisi terbaik dari semua partikel yang ada (*global*) pada dimensi d, dan X_{id} merupakan posisi partikel saat itu.

Kecepatan tersebut akan digunakan untuk menentukan posisi baru dari partikel tersebut. Posisi baru dari partikel adalah posisi partikel pada saat itu ditambahkan dengan kecepatan geraknya. Sehingga dapat dikalkulasi dengan persamaan berikut :

PSO Clustering

Menurut (Merwe & Engelbrecht, 2003) dalam konteks *clustering*, suatu partikel tunggal merepresentasikan *centroid cluster* sebanyak N_c Sehingga untuk setiap partikel X_1, dapat direpresentasikan sebagai berikut :

$$X_1 = \underset{c}{\left(m_{i1}, \ldots, m_{ij}, \ldots, m_{iN}\right)} \quad (3)$$

dimana m_{ij} merupakan *centroid cluster* ke-*j* pada partikel ke-*I*, pada cluster C_{ij}. Dari representasi artikel di atas, maka dapat disimpulkan bahwa kumpulan (*swarm*) dari *p centroid* merepresentasikan sejumlah *clustering*, sebanyak jumlah partikel, pada data tertentu.

Fungsi obyektif dari suatu partikel dapat dihitung dengan menggunakan *quantization error* yang dinyatakan sebagai berikut :

$$f = \frac{\sum_{j=1}^{N_c} \frac{\sum_{\forall Z_p \in Z} d\left(Z_p, m_j\right)}{|Z|}}{N_c} \quad (4)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$

dimana, *Z* merupakan kumpulan data yang di *cluster*, dan *d* merupakan fungsi jarak *euclidean* yang dinyatakan dalam persamaan berikut :

$$d(Z_p, m_j) = \sqrt{\sum_{i=1}^{d_m} (Z_{pi} - m_{ji})^2} \quad (5)$$

dimana d_m merupakan dimensi dari data yang di *cluster*.

IV. ALGORITMA

Metode yang digunakan dalam melakukan segmentasi pada citra digital aksara jawa pada perangkat lunak ini adalah metode *clustering*. Algoritma clustering yang digunakan adalah *Particle Swarm Optimization (PSO).* Algoritma *PSO clustering* dapat dinyatakan sebagai *pseudocode* berikut:

```
1. Inisialisasi semua posisi, pBest dan
   Gbest dengan piksel acak dari citra
   serta kecepatan partikel dengan nilai
   nol.
2. For iterasi = 1 to iterasi maksimum
   do:
   a. For each partikel p do :
      i. For each cluster c do :
           1. For each dimensi d do :
              a. Update kecepatan p pada
                 cluster c pada dimensi d
              b. Update posisi p pada
                 cluster c pada dimensi d
     ii.Masukan data citra ke p.posisi
   b. For each partikel p do :
      i. If fitness(p.posisi) <
         fitness(p.pBest):
            1. p.pBest ← p.posisi
            2. Masukkan data citra ke
               p.pBest
            3. If fitness(p.pBest) <
               fitness(gBest):
               a. gBest ← p.pBest
               b. Masukkan data citra
                  ke gBest
```

Pararel PSO Clustering

PSO merupakan algoritma yang *inherently parallel*, yaitu algoritma yang memiliki sifat dasar paralel, bukan serial, sehingga mudah untuk diimplementasikan pada komputasi paralel. Hal ini disebabkan oleh karena antara satu partikel dengan partikel yang lain tidak saling berhubungan, sehingga pemrosesan suatu partikel tidak perlu menunggu selesainya pemrosesan partikel yang lain.

Implementasi PSO Clustering Naif

Implementasi PSO Clustering secara pararel ini adalah implementasi yang paling sederhana. Implementasi ini dapat dinyatakan dalam pseudocode sebagai berikut:

```
1. Inisialisasi semua posisi, pBest dan
   gBest dengan piksel acak dari citra
   serta kecepatan partikel dengan nilai
   nol.
2. For iterasi = 1 to iterasi maksimum
   do:
   a. Update kecepatan dan posisi semua
      partikel secara paralel pada
      device
   b. Update pBest semua partikel secara
      pararel pada device
   c. Copy pBest dari device ke host
   d. For each partikel p do
         i. If fitness (p.pBest) <
            fitness (gBest):
             1. gBest ← p.pBest
             2. Masukkan data citra ke
                gBest
   e. Copy gBest dari host ke device
```

Implementasi PSO Clustering Full Device

Pada implementasi PSO Clustering ini, semua fungsi dari PSO dikerjakan secara

pararel pada device. Perbedaan dari implementasi naif adalah penentuan nilai gBest dikerjakan di device, bukan di host. Implementasi dinyatakan dalam pseudocode berikut :

```
1. Inisialisasi semua posisi, pBest dan
   gBest dengan pixel acak dari citra
   serta kecepatan partikel dengan nilai
   nol.
2. For iterasi = 1 to iterasi maksimum
   do:
      a. Update kecepatan dan posisi
         semua partikel secara pararel
         pada device.
      b. Update pBest semua partikel
         secara pararel device.
      c. Idx ← indeks partikel dengan
         nilai fitness pBest terkecil.
      d. Update gBest dengan pBest pada
         indeks partikel Idx.
```

V. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan dengan komputer Processor Core i3-2350M, Memory RAM 4.0 GB, Video Card Onboard NVIDIA GeForce GT 620M 1GB.

Terdapat kedupuluh citra aksara jawa dasar (Aksara nglegéna) untuk masing masing implementasi (CPU, Naif, full device) yang dibagi menjadi 4 buah citra uji. Keempat citra digital tersebut diuji dengan nilai cluster sebanyak 2. Jumlah partikel 20, 100 dan 1000. Jumlah iterasi sebanyak 20, 40, dan 60. Keduapuluh citra aksara jawa dasar (Aksara nglegéna) yang digunakan sebagai bahan uji adalah empat citra berformat JPG dan memiliki ukuran 300 X 40 pixel, sehingga tiap citra berukuran 5184 data.

Dari hasil pengujian didapatkan bahwa implementasi segmentasi citra dengan menggunakan PSO secara pararel memberikan peningkatan performa pemrosesan yang signifikan dibandingkan dengan segmentasi citra dengan PSO yang tidak pararel.

Jika diketahui n adalah jumlah partikel dan i adalah jumlah iterasi, maka kompleksitas waktu *worst case* dari implementasi diatas adalah :

1. O(i*2n) pada implementasi CPU, karena terdapat 2 buah loop sebanyak n elemen di dalam loop i.

2. O(i*n) pada implementasi naif, karena loop yang digunakan untuk mengupdate partikel dipararelkan sehingga kompleksitas loop tersebut menjadi O(1) atau konstan.

3. O(i*lg n) pada implementasi full device, karena loop untuk mengupdate partikel dan mencari gBest dipararelkan

sehingga menjadi O(1) (konstan), dan untuk mencari nilai fitness terkecil digunakan algoritma parallel reduction yang memiliki kompleksitas O(lg n).

Dilihat dari kompleksitas waktu setiap implementasi diatas, maka secara teoritis, untuk jumlah partikel cukup banyak, implementasi *full device* adalah yang tercepat karena waktu pemrosesannya relatif terhadap logaritma basis dua dari jumlah partikel, diikuti oleh implementasi naif, dan iplementasi pada CPU adalah implementasi yang paling lambat.

Untuk jumlah partikel yang cukup sedikit (20 partikel), implementasi pada CPU adalah implementasi yang berjalan paling cepat. Hal itu disebabkan karena peningkatan kecepatan kecepatannya tertutupi oleh overhead pada bagian lain, misalnya inisialisasi memori, *copy data* dan inisialisasi kernel.

Pada jumlah partikel yang cukup bannyak (100 dan 1000 partikel), implementasi pada GPU berjalan lebih cepat daripada implementasi pada CPU karena overhead pada pemrosesan pararel pada CUDA tidak signifikan dibandingkan peningkatan pemrosesannya. Implementasi *full device* adalah implementasi yang paling cepat, yaitu hampir setengah dari waktu pemrosesan pada implementasi CPU, diikuti oleh implementasi naif.

VI. KESIMPULAN DAN SARAN

Dari hasil penelitian, dapat ditarik kesimpulan yaitu:

1. Perangkat lunak untuk segmentasi citra digital aksara jawa dengan PSO yang berjalan pada CPU dan GPU dengan CUDA telah berhasil dibangun.

2. Secara umum, segmentasi citra digital dengan PSO yang berjalan pada GPU berjalan lebih cepat dibandingkan dengan segmentasi citra digital dengan PSO yang berjalan pada CPU. Dengan percepatan pada segmentasi citra digital dengan PSO yang berjalan pada GPU sekitar dua kali lipat dibandingkan segmentasi citra digital dengan PSO yang berjalan pada CPU.

3. Kualitas hasil clustering dari algoritma PSO yang berjalan pada GPU sebanding dengan kualitas hasil

clustering dari algoritma PSO yang berjalan pada CPU.

DAFTAR PUSTAKA

Abraham, A., Das, S., & Roy, S. (2008). Swarm Intelligence Algorithms for Data Clustering. *SOFT COMPUTING FOR KNOWLEDGE DISCOVERY AND DATA MINING* .

Arps, B., & Supriyanto. (2002). Special report JAVANESE ON THE INTERNET. *Caraka, The Messenger. A Newsletter for Javanists* , 37-38.

Ba-Karait, N. O., & Shamsuddin, S. M. (2008). Handwritten Digits Recognition using Particle Swarm Optimization. *Second Asia International Conference on Modelling & Simulation Vol2 No08* .

Cheo, C.-Y., & Ye, F. (2004). Particle Swarm Optimization Algorithm and Its Application to Clustering Analysis. *Networking, Sensing and Control, 2004 IEEE International Conference on* .

Cui, X., & Potok, T. E. (2005). Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm. *Journal of Computer Sciences (Special Issue)* .

Cui, X., Potok, T. E., & Palathingal, P. (2005). Document Clustering using Particle Swarm Optimization. *Proceedings 2005 IEEE Swarm Intelligence Symposium* .

Kristiadi, A., Pranowo, & Mudjihartono. (2013). PARALLEL PARTICLE SWARM OPTIMIZATION FOR IMAGE SEGMENTATION. *SDIWC* .

LAI , C.-C. (2006). A Novel Image Segmentation Approach Based on Particle Swarm Optimization. *IEICE TRANS. FUNDAMENTALS Vol E89 No1* .

Lin, C.-J., Wang, J.-G., & Lee, C.-Y. (2009). Pattern recognition using neural-fuzzy networks based on improved particle swam optimization. *Expert Systems with Applications Vol36* .

Merwe, D. v., & Engelbrecht, A. (2003). Data Clustering using Particle Swarm Optimization. *Evolutionary Computation, 2003. CEC '03* .

Mohamed, N. (2001). Aksara Jawi: Makna dan Fungsi. *Sari* , 121-131.

Mohsen, F. M., Hadhoud, M. M., & Amin, K. (2011). A new Optimization-Based Image Segmentation method By Particle Swarm Optimization. *(IJACSA) International Journal of Advanced Computer Science and Applications, Special Issue on Image Processing and Analysis* .

Murugesan, K., & Palaniswami, D. (2010). EFFICIENT COLOUR IMAGE SEGMENTATION USING MULTI-ELITIST- EXPONENTIAL PARTICLE SWARM OPTIMIZATION. *Journal of Theoretical and Applied Information Technology Vol 18 No1* .

Mussi, L., Cagnoni, S., & Daolio, F. (2009). GPU Based Road Sign Detection using Particle Swarm Optimization. *Intelligent Systems Design and Applications, ISDA '09. Vol9 No 9* .

Omran, M. (2004). Particle Swarm Optimization Methods for Pattern Recognition and Image Processing. *PhD Thesis Doctor in the Faculty of Engineering, Built Environment and Information Technology, University of Pretoria, November 2004* .

Omran, M., Engelbrecht, A., & Salman, A. (2005). Dynamic Clustering using Particle Swarm Optimization with Application in Unsupervised Image Classification. *Proc. 5th World Enformatika Conf. (ICCI)* .

Santosa, B., & Ningrum, M. (2009). Cat Swarm Optimization for Clustering. *IEEE International Conference of Soft Computing and Pattern Recognition* .

Widiarti, A. (2007). SEGMENTASI CITRA DOKUMEN TEKS SASTRA JAWA MODERN MEMPERGUNAKAN PROFIL PROYEKSI. *SIGMA, Vol. 10, No. 2, Juli 2007* , 167-176.

Ye, F., & Chen, C.-Y. (2005). Alternative KPSO-Clustering Algorithm. *Tamkang Journal of Science and Engineering Vol 6 No 5* .

Yih, J.-M., Lin, Y.-H., & Liu, H.-C. (2008). Clustering Analysis Method based on Fuzzy C-Means Algorithm of PSO and PPSO with Application in Image Data. *Proceedings of The 8th WSEAS International* .

Yih, J.-M., Lin, Y.-H., & Liu, H.-C. (2007). Clustering Analysis Method based on Fuzzy C-Means Algorithm of

PSO and PPSO with Application in Real Data. *INTERNATIONAL JOURNAL OF GEOLOGY vol4 no1* .